RECEIVED
CENTRAL FAX CENTER

MAR 19 2010

42390P9770 PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	
Neiger, et al	Examiner: Pyzocha, Michael J.
Serial No.: 09/752,134	Art Unit: 2137
Filing Date: December 27, 2000	
For: NEW PROCESSOR MODE FOR LIMITING THE OPERATION OF GUEST SOFTWARE RUNNING ON A VIRTUAL MACHINE SUPPORTED BY A VIRTUAL MACHINE MONITOR	

CERTIFICATE OF MAILING/TRANSMISSION

I hereby certify that this correspondence is being facsimile transmitted to the United States Patent and Trademark Office on the date indicated below and that this paper has been addressed to Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, fax number 571-273-8300.

Date of Deposit: March 19. 2010
Name of Person Transmitting Correspondence:

3/19/10

Signature Date

Mail Stop Appeal Brief Patents
Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

42390P9770 PATENT

TABLE OF CONTENTS

REAL PARTY IN INTEREST

The real party in interest is the assignee Intel Corporation.

RELATED APPEAL AND INTERFERENCES

None.

STATUS OF CLAIMS

Claims 9 and 31-40 (Rejected).

Claims 9 and 31-40 are rejected and are the subject of this Appeal Brief.

STATUS OF AMENDMENTS

All amendments have been entered.

SUMMARY OF CLAIMED SUBJECT MATTER

In the following discussion, the independent claims are read on one of many possible embodiments without limiting the claims.

9. A method (specification page 16, lines 21-23; figure 6) comprising:
   running guest software in a processor mode that enables the guest software to operate at a privilege level intended by the guest software (specification page 11, lines 1-3);
   identifying, within said processor mode, an attempt of the guest software to perform an operation restricted by said processor mode (specification page 16, lines 23-24; figure 6, 604);
   determining, within said processor mode, whether the attempt of the guest software would fail if the guest software was running outside said processor mode (specification page 17, lines 1-2, 6-11; figure 6, 606);
   allowing the guest software to attempt the operation within said processor mode in response to determining that the attempt would fail if the guest software was running outside said processor mode (specification page 17, lines 4-5, 12-13; figure 6, 610); and
   transferring control over the operation to an operating system running within said processor mode in response to the guest software attempting the operation (specification page 17, lines 2-3, 11-12; figure 6, 608).

33. A processor (specification page 20, lines 12-14; figure 9, 920) comprising:
   a storage location to store an indicator to indicate whether the processor is configured to execute guest software in a mode that enables the guest software to operate at a privilege level intended by the guest software (specification page 11, lines 15-19); and
   logic to execute the guest software in said processor mode (specification page 11, lines 1-3), to identify, within said processor mode, an attempt of the guest software to perform an operation restricted by said processor mode

(specification page 16, lines 23-24; figure 6, 604), to determine, within said processor mode, whether the attempt would fail if the guest software was running outside said processor mode (specification page 17, lines 1-2, 6-11; figure 6, 606), to allow the guest software to attempt the operation within said processor mode in response to determining that the attempt would fail if the guest software was running outside said processor mode (specification page 17, lines 4-5, 12-13; figure 6, 610), and to transfer control over the operation to an operating system running within said processor mode in response to the attempt (specification page 17, lines 2-3, 11-12; figure 6, 608).

36. A system (specification page 20, lines 15-17; figure 9, 900) comprising:

a memory (specification page 20, lines 18-22; figure 9, 930) to store application software and an operating system; and

a processor (specification page 20, lines 12-14; figure 9, 920) to execute the application software in a processor mode that enables the application software to operate at a privilege level intended by the application software (specification page 11, lines 1-3), to identify, within said processor mode, an attempt of the application software to perform an operation restricted by said processor mode (specification page 16, lines 23-24; figure 6, 604), to determine, within said processor mode, whether the attempt would fail if the application software was running outside said processor mode (specification page 17, lines 1-2, 6-11; figure 6, 606), to allow the application software to attempt the operation within said processor mode in response to determining that the attempt would fail if the guest software was running outside said processor mode (specification page 17, lines 4-5, 12-13; figure 6, 610), and to transfer control over the operation to the operating system running within said processor mode in response to the attempt (specification page 17, lines 2-3, 11-12; figure 6, 608).

39. A computer readable storage medium (specification page 5, lines 14-21) that provides instructions, which when executed on a processor, cause the processor to:

present a virtual machine to guest software in a processor mode that enables the guest software to operate at a privilege level intended by the guest software (specification page 11, lines 1-3); and

handle a virtualization trap (specification page 11, lines 10-14), where the virtualization trap is generated (specification page 17, lines 2-3, 11-12; figure 6, 608) in response to an attempt (specification page 16, lines 23-24; figure 6, 604) of the guest software to perform an operation restricted by said processor mode and a determination (specification page 17, lines 1-2, 6-11; figure 6, 606), within said processor mode, that the attempt would succeed if the guest software was running outside the virtual machine.

At this point, no issue has been raised that would suggest that the words in the claims have any meanings other than their ordinary meanings. Nothing in this section should be taken as an indication that any claim term has a meaning other than its ordinary meaning.

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

A. Whether claims 9 and 31-40 comply with the written description requirement of 35 U.S.C. §112, first paragraph.

B. Whether the specification provides proper antecedent basis for claims 9 and 31-40.

C. Whether claims 9 and 31-40 are unpatentable under 35 U.S.C. §103(a) over U.S. Patent No. 5,522,075 ("Robinson") in view of U.S. Patent Publication No. 2002/0069335 ("Flynn").

ARGUMENT

A. Do claims 9 and 31-40 comply with the written description requirement?

The examiner incorrectly argues that the limitation that the determining is performed within the processor mode is not supported by the specification. However, Figure 6 and the description of method 600 in the specification (line 21 of page 16 to line 10 of page 18) make it clear that the determining is performed within the processor mode for the reasons set forth below.

As explained on page 11, lines 1-3 of the specification, guest software runs within the processor mode (e.g., V32 mode) that allows the guest software to run at its intended privilege level. Therefore, method 600 begins within the processor mode, for the specification, on page 16, lines 23-24, state that method 600 begins with identifying an attempt of guest software to perform an operation that may be restricted by V32 mode.

The specification also explains that method 600 is a method for generating virtualization traps, page 16, lines 21-22, which correspond to, as described on page 12, lines 4-6, transitions out of V32 mode. Figure 600 and the description of method 600 clearly describe the condition (the "Y" branch from block 606) that causes a virtualization trap (in block 608). As shown in Figure 600, the determination of whether the attempt of the guest software would fail if the guest software was running outside of the processor mode (block 606) conditionally results in a virtualization trap (block 608), therefore the determination must occur within the processor mode or else it would not be possible to transition out of the processor mode.

Therefore, the specification clearly describes that the determining is performed within the processor mode.

B. Does the specification provide proper antecedent basis for the claims?

For the reasons set forth in Section A, the specification provides proper antecedent basis for the claims.

C. Are claims 9 and 31-40 patentable over Robinson and Flylnn?

Independent claim 9 requires that the determining is performed within the processor mode. Independent claims 33, 36, and 39 include corresponding limitations. Each of dependent claims 31, 32, 34, 35, 37, and 38 include this limitation based on its dependence on a claim including this limitation.

None of Robinson, Flylnn, or the combination of Robinson and Flylnn describe or suggest this limitation. The examiner incorrectly cites paragraphs 0009 and 0010 of Flylnn as describing the determining. However, in lines 1-3 of paragraph 0010, Flylnn describes the control program gaining control for the determining, which, according to the examiner's argument that Flylnn's control program corresponds to the virtual machine monitor of the present invention, is the opposite of the determining within the processor mode because the virtual machine monitor of the present invention runs outside the processor mode. Therefore, Flylnn teaches away from the required limitation of determining within the processor mode.

* * *

Applicant respectfully requests that each of the final rejections be reversed and that the claims subject to this Appeal be allowed to issue. Please charge any necessary fees, including extension fees, to our Deposit Account No. 50-0221.

Respectfully submitted,

Date: March 19, 2010

Thomas R. Lane
Registration No. 42,781

CLAIMS APPENDIX

The claims on appeal are:

9. A method comprising:

running guest software in a processor mode that enables the guest software to operate at a privilege level intended by the guest software;

identifying, within said processor mode, an attempt of the guest software to perform an operation restricted by said processor mode;

determining, within said processor mode, whether the attempt of the guest software would fail if the guest software was running outside said processor mode;

allowing the guest software to attempt the operation within said processor mode in response to determining that the attempt would fail if the guest software was running outside said processor mode; and

transferring control over the operation to an operating system running within said processor mode in response to the guest software attempting the operation.

31. The method of claim 9 wherein determining that the attempt of the guest software would fail includes determining that the guest software is running with insufficient privilege to perform the operation.

32. The method of claim 9 further comprising exiting said processor mode to transfer control over the operation to a virtual machine monitor running outside said processor mode in response to determining that the attempt would succeed if the guest software was running outside said processor mode.

33. A processor comprising:

a storage location to store an indicator to indicate whether the processor is configured to execute guest software in a mode that enables the guest software to operate at a privilege level intended by the guest software; and

logic to execute the guest software in said processor mode, to identify, within said processor mode, an attempt of the guest software to perform an operation restricted by said processor mode, to determine, within said processor mode, whether the attempt would fail if the guest software was running outside said processor mode, to allow the guest software to attempt the operation within said processor mode in response to determining that the attempt would fail if the guest software was running outside said processor mode, and to transfer control over the operation to an operating system running within said processor mode in response to the attempt.

34. The processor of claim 33 wherein the logic is to determine whether the attempt would fail by determining whether the guest software is running with sufficient privilege to perform the operation.

35. The processor of claim 33 wherein the logic is also to exit said processor mode to transfer control over the operation to a virtual machine monitor running outside said processor mode in response to determining that the attempt would succeed if the guest software was running outside said processor mode.

36. A system comprising:

a memory to store application software and an operating system; and

a processor to execute the application software in a processor mode that enables the application software to operate at a privilege level intended by the application software, to identify, within said processor mode, an attempt of the application software to perform an operation restricted by said processor mode, to determine, within said processor mode, whether the attempt would fail if the application software was running outside said processor mode, to allow the application software to attempt the operation within said processor mode in response to determining that the attempt would fail if the guest software was running outside said processor mode, and to transfer control

over the operation to the operating system running within said processor mode in response to the attempt.

37. The system of claim 36 wherein the processor is to determine whether the attempt would fail by determining whether the application software is running with sufficient privilege to perform the operation.

38. The system of claim 36 wherein:

the memory is also to store a virtual machine monitor; and

the processor is also to exit said processor mode to transfer control over the operation to the virtual machine monitor running outside said processor mode in response to determining that the attempt would succeed.

39. A computer readable storage medium that provides instructions, which when executed on a processor, cause the processor to:

present a virtual machine to guest software in a processor mode that enables the guest software to operate at a privilege level intended by the guest software; and

handle a virtualization trap, where the virtualization trap is generated in response to an attempt of the guest software to perform an operation restricted by said processor mode and a determination, within said processor mode, that the attempt would succeed if the guest software was running outside the virtual machine.

40. The computer readable storage medium of claim 39 wherein the determination that the attempt would succeed includes determining whether the application software is running with sufficient privilege to perform the operation.

EVIDENCE APPENDIX

None.

42390P9770 PATENT

RELATED PROCEEDINGS APPENDIX

None.